

Infrastructure Modeling and Optimization to Solve Real-time Railway Traffic Management Problems

László Lindenmaier^{1*}, István Ferenc Lövétei¹, Gábor Lukács¹, Szilárd Aradi¹

¹ Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, H-1111 Budapest, 2 Stoczek street, Hungary

* Corresponding author, e-mail: lindenmaier.laszlo@kjk.bme.hu

Received: 17 May 2021, Accepted: 15 June 2021, Published online: 19 August 2021

Abstract

Rail transportation helps to reach the global climate targets because it is characterized by low emission. The passenger and freight volumes on the railway increase yearly in line with EU targets. However, delays of passenger and freight trains decrease the punctuality and the reliability of the railway sector and the development of the infrastructure is not enough to increase the average speed of trains. Delays mean cost to the passengers, railway operators, infrastructure managers, and all railway undertakers. Therefore, the reason for the most significant optimization target is to minimize delays. In this paper, a possible solution has been described to solve the real-time railway traffic management problems by applying a mixed-integer linear programming approach. For validation of the research result, one simplified case study has been presented. Based on the result, the presented solution can provide effective support to dispatchers in solving real-time traffic management problems.

Keywords

rescheduling, rerouting, mixed linear integer programming

This article was originally published with some errors. This version has been corrected/amended. Please see Corrigendum (<https://doi.org/10.3311/PPtr.19186>)

1 Introduction

Rail transportation helps to reach the global climate targets. It reduces the CO₂ and CHG emission and also the congestion on routes. The passenger and freight volumes on the railway increase year after year in line with EU targets described in the European Green Deal (European Commission, 2019a) and shown in Fig. 1. One key point of this agreement is the 'Initiatives to increase and better manage the capacity of railways and inland waterways' (European Commission, 2019b).

The main target of the technical pillar of the fourth railway package is to boost the competitiveness of the railway sector.

Delays of passenger and freight trains decrease the punctuality and the reliability of the railway sector. The average timetabled speed of freight services is 45.63 km/h for domestic freight trains and 45.70 km/h for international freight trains (see Fig. 2).

Expanding the infrastructure is expensive, and in most cases, it is not enough to increase the average speed of (freight) trains. The better use of the limited infrastructure can allow more vehicles to travel on the network with decreased journey time. In this case, there is no need to make a high investment in the infrastructure to increase the railway traffic efficiency.

Different countries/railways apply different priority rules that determine the train sequence on the infrastructure because some trains have higher priority than others due to the public service operation (PSO). The applied priority rules can be crucial, especially in disturbance situations. The most common principle is the first-in-first-out (FIFO) principle. It means that the first arrived train has the right to go through the bottleneck, even if its maximum velocity is lower than that of the follower train. In addition to priority principles, the infrastructure operators also determine several dispatching rules based on other agreements.

In the operational controlling process, dispatchers have to make the decision based on priority rules and agreements mentioned above. Nowadays, they rarely use decision-support systems to find the optimal result, and they work only based on their experiences using priority rules and agreements. The reason of ignoring decision-support systems is their high computation time. In other words, the necessary suggestions for decisions are available too late for dispatchers. The slowness of the current decision support systems is due to the fact that to make suitable suggestions for dispatchers, several parameters have to be

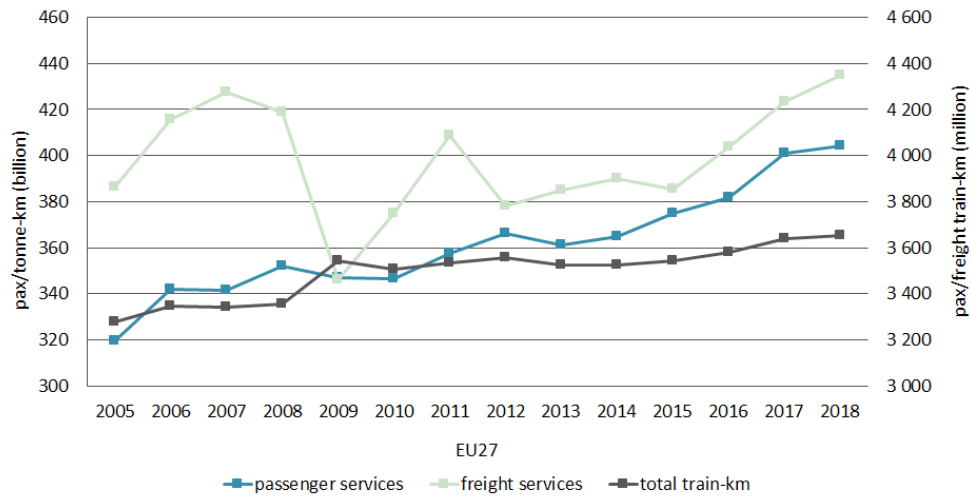


Fig. 1 Passenger and freight volumes, EU27 (European Commission, 2021)

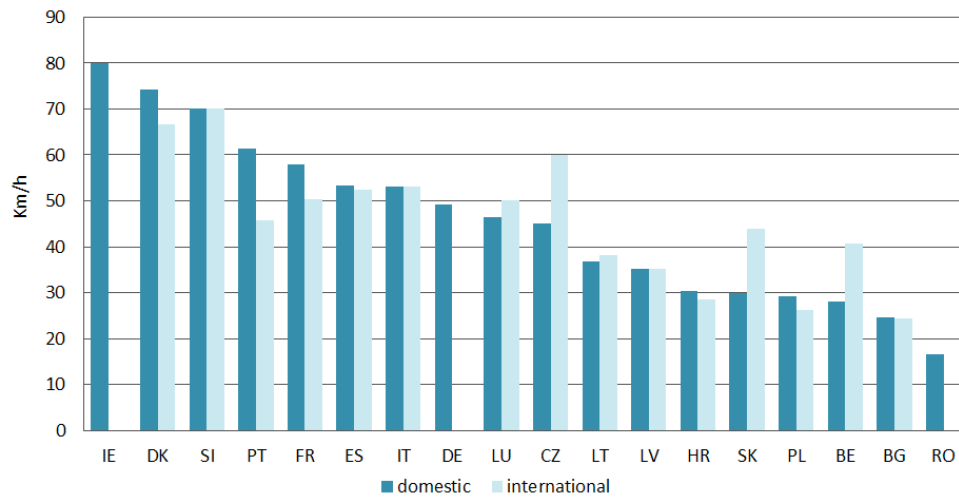


Fig. 2 Average timetabled speed for freight services (European Commission, 2021)

considered that can result in huge state space. Of course, the state space continues to grow with the size of the considered infrastructure. There are several open issues in this area, such as optimization options, optimization algorithms, and so on (see Section 1.2).

The decision-making process is crucial in the case of traffic disturbances. A disturbance leads to a primary delay to a train which can cause secondary delays to other trains. It is possible to identify the disturbances that cause the primary delay, but no action can be taken against them. Therefore, in such situations, the main target is to minimize the total secondary delay, i.e. to minimize the sum of the secondary delays of affected trains. The decision-making algorithms can help to solve these situations in an effective way by finding the optimal solution. The optimization target can also be complicated, i.e. in addition to minimizing the secondary delay, additional or even different targets may appear (e.g. energy-efficiency, etc.).

1.1 Motivation

Delays mean cost to the passengers, railway operators, infrastructure managers, and all railway undertakers. That is why the most significant optimization target is to minimize it. In the first assumption, the task is to perform a route search task (rerouting) in a given part of the railway network considering the safety rules based on the 'Blocking time theory' (Pachl, 2014). This theory contains every safe time to form and release a route as used in this paper. The critical points, (like bottlenecks) are the switches and single-track sections. Two vehicles may not pass through these sections at the same time. Dispatchers have to determine the train ordering (rescheduling and reordering). This solution has to be feasible. To solve this problem based on the mathematical way, it is needed to find efficacious algorithms which provide a feasible solution. During the decision-making process, it is possible to apply other complex traffic control rules beyond the FIFO principle. For example, energy efficiency also can be examined.

The modeling of the infrastructure is an essential part of traffic management problems. To receive the best results, microscopic models are needed. The microscopic model means, that the smallest modeled unit of the infrastructure is track-circuit. The track-circuit is the safety element of a track section with relevant properties, like maximum allowed speed, length, and gradient. The decision-making algorithm has to use these parameters. The generalized representation of the railway infrastructure is usually a directed graph described by matrices. The size of the matrices grows proportionally with the size of the infrastructure.

Another significant part of the system is the rolling stock which has essential properties like acceleration, braking ability, length, entry- and exit point for a given infrastructure. Trains can be stopped at a predefined platform to allow passenger exchange, the time of which is also predetermined. Of course, every train has an own timetable.

Mixed Integer Linear Programming (MILP) is a tool to solve mathematically the real-time railway traffic management problem (rt-RTMP) regarding the infrastructure model, the train model, and traffic rules. In this paper, the authors use an extended MILP formulation to show a solution for the rt-RTMP. The algorithm determines the best route for every train taking into consideration the related constraints (described in Section 4). The main target is to find a feasible solution for the rt-RTMP problem in an effective way using MILP in a MATLAB environment.

1.2 Related work

Several solutions exist in the literature to the enhance energy consumption of the railway applications. In Novak et al. (2018) the authors use multiparametric quadratic programming to calculate the optimal train control law resulting in a time-varying piecewise affine function. In (Farooqi et al., 2008) the cooperative energy efficient train control problem is solved with switched Nonlinear Model Predictive Controller (NMPC), while in a further extension (Farooqi et al., 2009), at network level, a Dissension based Adaptive Law (DAL) is then proposed to adjust the parameters of the NMPC cost so as to efficiently share the available regenerated braking energy. The authors of (Abels et al., 2019) propose a solution for real-world train scheduling with hybrid Answer Set Programming (ASP). Their approach allows to specifically account for the different types of constraints induced by routing, scheduling, and optimization. In (Toletti et al., 2016) the authors design an enhanced version of the Resource Conflict Graph (RCG) approach

for railway rescheduling that considers the reduction of tractive energy consumption in the rescheduling process. In Pellegrini et al. (2012), the authors detail a possible solution to the rt-RTMP problem using MILP to minimize secondary delays in a case study that describes unexpected events which perturb the operations. They later developed this solution with additions considering the efficiency of the algorithm and automatic generation of the configuration space (Pellegrini et al., 2015). In this paper, we present the proposed solution by Pellegrini et al. with one MATLAB implementation. The results we have achieved are described in a case study.

2 Infrastructure modeling

For resolving the real-time railway traffic management problem an appropriate and detailed infrastructure model is needed. Pellegrini et al. (2012) used a track-circuit based infrastructure model in their optimization algorithm. Track-circuits are the fundamental elements of railway infrastructure that help to automatically detect the presence of the train in a given section. Although this model considers the possible order of the track circuits, it does not contain rules for defining the preceding and subsequent track-circuits. This approach can lead to a very insufficient and slow infrastructure modeling process because the order of track-circuits on a given route must be defined by the user.

The main objective of our research was to extend the infrastructure model described by Pellegrini et al. (2012) to design a more effective modeling process. The infrastructure model contains four layers demonstrated by Fig. 3.

The lowermost layer of the infrastructure model is a graph-based description. In this layer, the base components are the graph nodes connected by undirected edges. There are several advantages of extending the model described by Pellegrini et al. (2012) with a graph layer. Among others, such an advantage is that the connections of the track-circuits can be automatically generated and the signals can be assigned to the nodes of the graph.

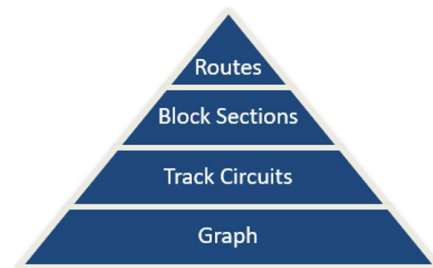


Fig. 3 Layer structure of the infrastructure model

The track-circuits are sets of edges, so this representation enables modeling railway switches. A block section consists of track-circuits delimited by main signals. A set of block sections constructs a route connecting two terminal nodes of the graph. The terminal nodes (the nodes on the border of the infrastructure and the nodes which delaminate? platforms) represent the places where trains can start and finish their movement. This infrastructure model describes four layers which are shown in Fig. 4.

Fig. 4 shows a case study of the described infrastructure model. The nodes on the border of the modeled infrastructure are entry or exit points of the control area. The modeled infrastructure contains two platforms to handle the departing and arriving trains. The terminal nodes, including the border nodes of the platforms, are represented by triangles in Fig. 4. The graph model of the infrastructure consists of 25 edges that cover 13 track-circuits and 6 railway switches. The signals delimiting the block sections are also shown in Fig. 4. It is important to note that block sections are not necessarily the same in different directions. For example, on the route connecting tc1 and tc12 through tc6, there are four block sections from tc1 to tc12: tc1, tc3-tc6, tc9-tc11, and tc12. However, from tc12 to tc1 there are only three block sections: tc12, tc11-tc6, and tc4-tc1.

As has already been discussed, the routes consist of block sections between two terminal nodes. There are 12 routes between entry and exit points in this infrastructure model, and 8 routes have a start or endpoint on a platform. The platforms are represented with dashed lines in Fig. 4. Since track-circuits may have different properties depending on route choice and direction, several nominal track-circuits are generated based on the real track-circuits. From each real track-circuit, two nominal track-circuits (for both directions) are generated as many times as the routes contain the corresponding real track-circuit. Furthermore, two dummy track-circuits are also defined: tc_0 and tc_∞ . These represent the entry and

exit locations of the control area. In complex infrastructure, the number of nominal track-circuits can be very high, which requires high computational resources for the optimization algorithm. In the following, nominal track-circuits will be referred to as track-circuits.

By extending the track-circuit-based infrastructure model with a graph-based descriptor, the upper layers can be automatically generated. For the construction of a graph-based model, a complete Matlab toolchain has been implemented with a GUI, shown in Fig. 5.

To generate the upper layer of the infrastructure model automatically, some properties should be assigned to nodes and edges of the graph. For example, automatic block section generation cannot be performed without assigning signals to the corresponding nodes. Table 1 summarizes these properties.

The node type is automatically defined based on the track-circuit definition. If a track-circuit consists of one edge of the graph, the nodes delimiting the edge are normal nodes. If the track-circuit consists of multiple edges, the intermediate node is stored as a hidden node modeling a railway switch. A node is considered a terminal node if it is connected to only one edge or either to a platform container edge. Signals are assigned to nodes where they are applied, i.e. the corresponding node has a signal location property. The signaled node from which the signal is valid for traveling trains must be also selected to define the direction. This node has a signal direction property. The length and maximum velocity property of the edges of the model are important input parameters. The platform indicator supports the generation of inner terminal nodes.

In the actual Matlab toolchain, the user defines the routes as follows: first, the user selects the corresponding track-circuits; however, route validity is automatically investigated. A route is valid when the selected track-circuits connect exactly two terminal nodes. Besides, the graph model allows the implementation of automatic route

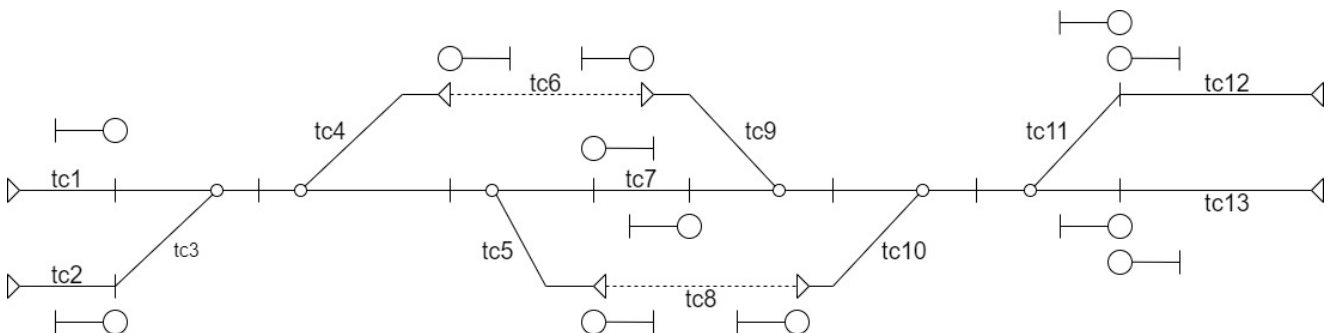


Fig. 4 Infrastructure model

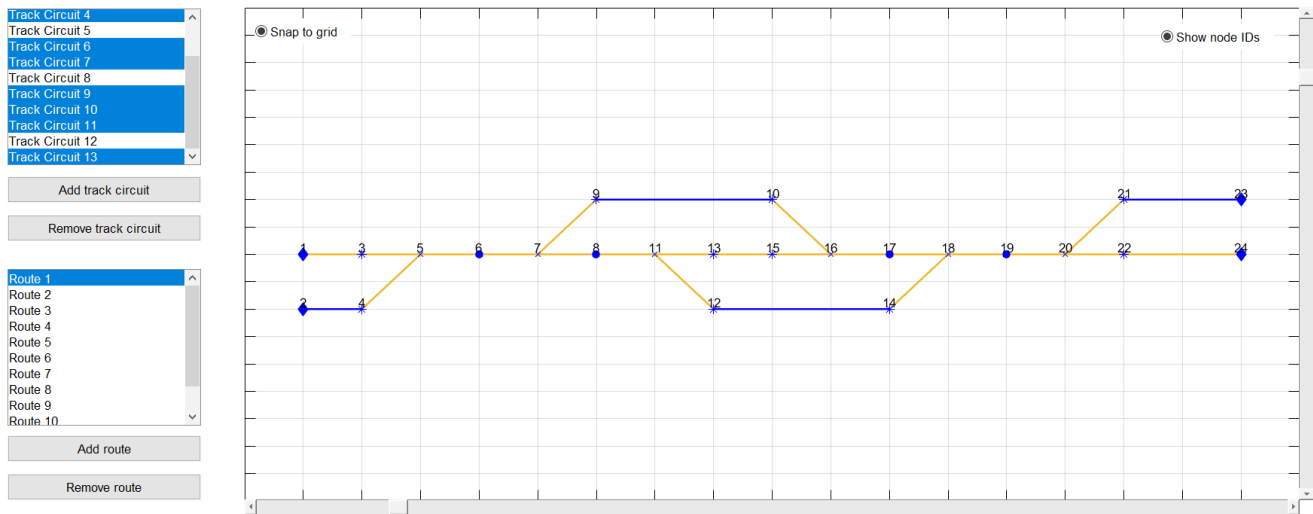


Fig. 5 Matlab GUI for infrastructure modeling

Table 1 Graph properties

Node properties	
Node type	Normal/Hidden/Terminal
Signal(s)	Location/Direction
Edge properties	
Edge length	Total length of represented track
Maximum velocity	Maximum velocity of the train on the corresponding edge
Platform indicator	Indicates whether the edge contains platform

definition based on iterative graph search algorithms (like breadth-first search – (Kozen, 1992) or Dijkstra algorithm – (Dijkstra 1959)) extended by application-dependent rules. For example, suppose the drawn graph represents the real infrastructure. If the drawn graph represents the shape of the real infrastructure, the shunting movements could be neglected by comparing the total turn angle of the movement to a well-defined threshold. Let us assume a 90° threshold. In this case, tc12-tc13, tc1-tc2, and tc6-tc8 (see Fig. 4) routes could be easily filtered.

3 Railway traffic model

Modeling and optimizing railway traffic transportation requires a railway traffic model. This paper describes a simple railway traffic model using a constant velocity model. In practice, an essential input of railway traffic modeling is the preplanned timetable. During the timetable planning process, the elements of the existing infrastructure are also taken into account. Several parameters must be given by the user that are summed in Table 2. In real railway traffic, some trains are in connection with

Table 2 Train properties

Property	Sign and meaning
Motion status	$MoS_t = \begin{cases} 1, & \text{forward moving} \\ -1, & \text{shunting} \\ 0, & \text{standstill} \end{cases}$
Platform indicator	$PL_t = \begin{cases} 0, & \text{passing through} \\ 1, & \text{departs from platform} \\ 2, & \text{terminates at platform} \end{cases}$
Maximum velocity	$v_{max,t}$
Train length	L_t
Scheduled entry time	$init_t$
Scheduled exit time	$exit_t$
Start node	Terminal node from where the train starts the route
Exit node	Terminal node where the train finishes the route

another train, some of them result from the turn-around, join or split of other trains. This process is handled by some train model parameters summarized in Table 3.

The spatial coherence of the trains has to be ensured next to time coherence (see in Subsection 4.2) with the definition of connections and changes in the configuration of rolling stock of the trains. For example, train t could connect to train i if train t terminates at platform. In such case the start node of train t is automatically set to the corresponding platform. This requirement is applied for rolling stock configuration change as well. In Matlab infrastructure designer GUI, when a new train is defined, only those trains can be selected for connection or change of the rolling stock configuration that terminate at a platform. The spatial coherence requirement is automatically established in this model.

Table 3 Configuration parameters of rolling stock

Parameter	Sign and meaning
Connection of trains	$C_t(t) = \begin{cases} t', & \text{if connection with } t' \\ [] & \text{no connection} \end{cases}$
Join/split/turn-around of trains	$I_t(t) = \begin{cases} t', & \text{if train } t \text{ results from train } t' \\ [] & \text{no preceding train} \end{cases}$
Use of the same rolling stock	$RoS_t(t) = \begin{cases} t', & \text{using rolling stock of train } t' \\ [] & \text{no train with the same rolling stock} \end{cases}$

An important parameter of optimization is the running time on track-circuit tc . The infrastructure model is prepared for different running times on a given real track-circuit used by different routes. Although, most of the previous research on this topic does not handle train-dependent running time. This paper introduces a simple method to resolve this issue. The complexity of the algorithm increases with the infrastructure's size, considering a real velocity profile would not satisfy the real-time traffic management requirements. The train velocity is a discrete variable in this model. The vehicles can have three motion states: forward-moving, shunting, and standstill. The moving speed is the minimum value of the maximum velocity allowed on the current track-circuit and the maximum velocity allowed of the train. Thus, the velocity of a train is given by Eq. (1) depending on the motion status, $v_{max,tc}$ maximum velocity allowed on the track-circuit, and $v_{max,t}$ maximum velocity of the train.

$$v_{t,tc} = \begin{cases} 0, & \text{if } MoS_t = 0 \\ \min(v_{max,tc}, v_{max,t}), & \text{otherwise} \end{cases} \quad (1)$$

Based on the $v_{t,tc}$ train velocity, the $run_{t,tc}$ train model-dependent running time on a given tc track-circuit is computed by Eq. (2).

$$run_{t,tc} = \frac{L_{tc}}{v_{t,tc}} \alpha(r^{tc}) \quad (2)$$

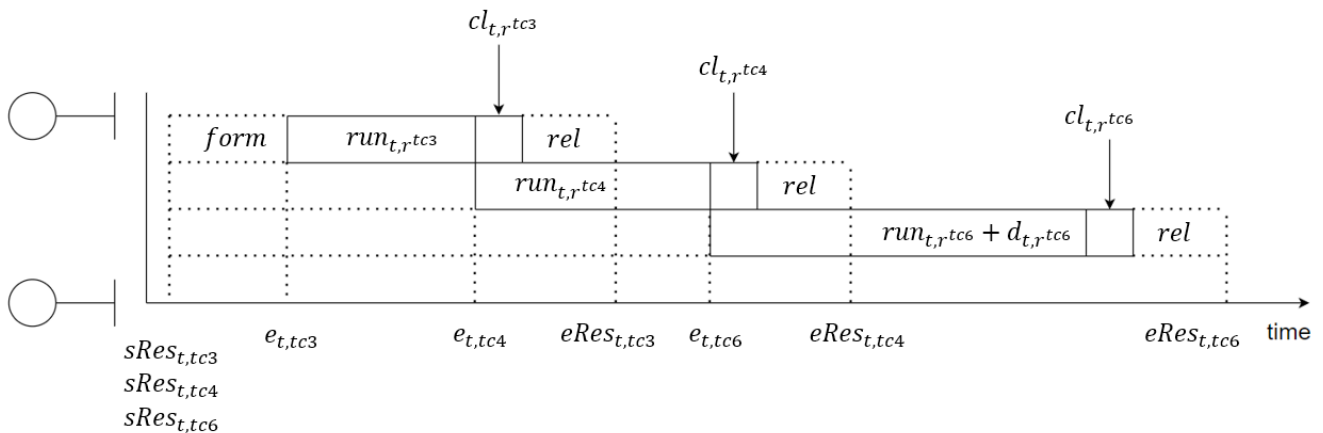
In Eq. (2) $\alpha(r^{tc})$ denotes the route-dependent running time factor on the route r^{tc} , which contains the track-circuit tc . This factor lets one consider route-dependent running time due to several environmental circumstances (i.e., topographic conditions). In this research, this factor was neglected, i.e. set to 1 for simplicity.

It is important to note that this railway model established the rerouting optimization principle. The train model does not include a specific route, but defines the starting and end node of the train. Thus, the optimization algorithm can define the set of available routes between the corresponding terminal nodes and choose the optimal route taking into account the objective function i.e. the global minimum secondary delay.

The constant velocity model is straightforward; however, the dynamic limits of trains are neglected (infinity accelerations and decelerations are assumed), therefore applying additional safety times are suggested. The so-called *form* formation and *rel* release times are generic train independent parameters of railway management. It means that train reserves the block section formation time unit before it enters it, and it still reserves the release time units after it leaves it. It should also be mentioned that after the first axle of a train exits from a given track-circuit, the rest of the rolling stock configuration still occupies it until the $cl_{t,tc}$ clear time elapses. The clearing time could be easily computed based on the total length of the train L_t and the $v_{t,tc}$ velocity as in Eq. (3).

$$cl_{t,tc} = \frac{L_t}{v_{t,tc}} \alpha(r^{tc}) \quad (3)$$

The speed-distance diagram of train t on block section $tc3$ - $tc6$ (see Fig. 4) on route r is demonstrated in Fig. 6.


Fig. 6 The time-distance diagram of a train on a block section (Pellegrini et al., 2012)

In Fig. 6, $sRes_{t,tc}$ and $eRes_{t,tc}$ denote the time when train starts and finishes the reservation of track-circuit tc . The $d_{t,tc}$ is the delay assigned to train t at track-circuit tc . This variable is the key to schedule traffic participants. Delays can only be assigned to signals (at the end of block sections). It should be noted that there are different signaling systems. In general, in an n aspect system, there are $n - 2$ block sections for braking. This research considered two-aspect systems which is shown in the time-distance diagram represented by Fig. 4.

4 Mixed-integer linear programming optimization

Linear programming is a common solution for optimization problems in which the objective function and the constraints are described by linear functions. Mixed-integer linear programming is a subset of general linear programming where some of the variables are integers. This section describes the detailed MILP formula used for the optimal solution of rt-RTMP. Subsection 4.1 details the objective function and the state variables, while Subsection 4.2 explains the formulae of the constraints. The constant input parameters of the algorithm are given in Table 4.

4.1 State variables and objective function

The state vector of the system used for the definition of constraints and objective function consists of several state variables. The elements of the state vector are defined according to the linear constraints. The MILP algorithm searches for values of the state variables to obtain the minimum of the objective function. Table 5 shows the elements of the state vector.

The state vector is constructed based on the system state using Matlab optimization algorithm. The system state is described in a matrix format. The number of rows of the state matrix corresponds to the number of trains assigned to the infrastructure. The number of columns equals the number of state variables of a given train (e.g. entry times to the nominal track-circuits, delays assigned at nominal track-circuits, etc.). Eq. (4) represents this state matrix.

$$X = \begin{bmatrix} e_{1,tc} & d_{1,tc} & sRes_{1,rtc} & eRes_{1,rtc} & x_{1,r} & y_{1,t',rtc} \\ e_{2,tc} & d_{2,tc} & sRes_{2,rtc} & eRes_{2,rtc} & x_{2,r} & y_{2,t',rtc} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ e_{n,tc} & d_{n,tc} & sRes_{n,rtc} & eRes_{n,rtc} & x_{n,r} & y_{n,t',rtc} \end{bmatrix} \quad (4)$$

In Eq. (4) $n = t \vee$ denotes the number of trains assigned to the infrastructure. The state vector is constructed by reshaping the state matrix to a column vector done by Eq. (5), where X_i denotes the i -th row of the X state matrix.

Table 4 Input parameters (constants) of MILP optimization

Parameter	Meaning
T, R	set of trains and routes
RTC, TC	set of real and nominal track-circuits
RTC_r, TC_t	set of real and nominal track-circuits available for train t
MoS_t	motion status of train t
$PL \subset RTC$	set of real track-circuits containing a platform
$R_t \subseteq R$	set of routes available for train t
RCT^r	set of real track-circuits that correspond to route r
rtc^{tc}	real track-circuit that corresponds to nominal track-circuit tc
r^{tc}, bs_{tc}	route and block section that belong to tc
p_{tc}, stc	preceding and subsequent track-circuits of tc
$ebs(tc)$	indicator function ($tc \notin \{tc_0, tc_n\}$): 1 if the track-circuit tc belongs to a terminal (either the first or the last) block section on its route, 0 otherwise
ref_{tc}	reference track circuit for the reservation of tc ($tc \notin \{tc_0, tc_n\}$). In two-aspects signaling system it is the first track circuit of block section bs_{tc}
$run_{t,tc}, cl_{t,tc}$	running and clearing time of track-circuit tc
$form, rel$	formation and release time
$init_t, exit_t$	scheduled entry and exit time of train t
$I(t, t')$	indicator function: 1 if $I_t(t) = t'$, 0 otherwise
$C(t, t')$	indicator function: 1 if $C_t(t) = t'$, 0 otherwise
$RoS(t, t')$	indicator function: 1 if $RoS_t(t) = t'$, 0 otherwise
ms	minimum separation time between the arrival of a train and departure of another train using the same rolling stock or trains in connection
M	large constant

Table 5 State variables of MILP optimization

Variables	Meaning
$e_{t,tc}$	entry time of train t to track-circuit tc
$d_{t,tc}$	delay assigned by the optimization to train t at track-circuit tc
$sRes_{t,rtc}$	time when train t starts to reserve real-track-circuit rtc
$eRes_{t,rtc}$	time when train t ends reserving real-track-circuit rtc
$x_{t,r}$	binary indicator variable: 1 if train t uses route r , 0 otherwise
$y_{t,t',rtc}$	binary indicator variable: 1 if train t uses real track-circuit rtc before train t' , 0 otherwise

$$x = [X_1 \quad X_2 \quad \dots \quad X_n]^T \quad (5)$$

In real-time railway traffic management, the runtime of the optimization algorithm is crucial. The runtime of the MILP depends highly on the dimension of the optimized state. In the presented problem, the state dimension depends on the number of trains and the complexity of the infrastructure. The n_x number of elements in the optimized state is given by Eq. (6).

$$n_x = n \cdot (2n_{tc} + 2n_{rtc} + n_r + n \cdot n_{rtc} + 1) \quad (6)$$

In Eq. (6) n_{tc} , n_{rtc} and n_r denote the number of nominal track-circuits, real track-circuits, and routes, respectively. Based on Eq. (6), the runtime of the optimization increases with the number of assigned trains and the complexity of the infrastructure. It means that the real-time decision-making requirement could hardly be satisfied with low computational capacity in an infrastructure that consists of many real-track circuits and possible routes.

The previously described state vector is extended by the delays assigned to the trains. This step is needed for the definition of the objective function. The D_t total delay of train t is defined as the difference between the time when the train exits the control area and the scheduled exit time. The D_t total delay of train t is computed as in Eq. (7).

$$D_t = e_{t,tc} - exit_t \quad (7)$$

The objective value optimized by the MILP algorithm is computed as the sum of the total delay of each train. The objective function is represented by Eq. (8).

$$\min \left(\sum_{t=1}^n D_t \right) \quad (8)$$

The objective function described in Eq. (8) considers the sum of the trains' delays. However, it is common to prioritize between trains. In this approach, it is easy to perform train priority based on predefined w_t train weight parameters. Thus, the objective function can be computed as in Eq. (9).

$$\min \left(\sum_{t=1}^n w_t D_t \right) \quad (9)$$

In this research, we applied equal weights for the trains due to simplicity.

4.2 Constraints

The constraints are formed according to the railway traffic management regulations. The constraints are given as linear functions of the predefined state vector to be able to insert them into the linear programming framework. According to previous research (Pellegrini et al., 2012) on this topic, four groups of constraints should be created: time concerning constraints, constraints for managing delays, constraints due to the change of rolling stock configuration, and capacity constraints.

The time constraints consider the chronological order of the track-circuits used by trains, the time coherence of connecting trains, and the coherence of the objective value. The time constraints are detailed in Table 6.

Table 6 Time concerning constraints

Constraint	Description
1	A train cannot enter the infrastructure before it is scheduled.
2	The entry time of track-circuits unused by the corresponding train equals 0.
3	A train cannot enter a track-circuit before leaving the preceding one.
4	A train can use exactly one route.
5	A minimum separation time must be granted between the arrival and departure of connecting trains.
6	The total delay of a train cannot be less than the difference between the exact and scheduled exit time.

The constraints for managing delays are intended to define the local delays assigned to trains at signals due to collision avoidance. The managing of delays is also based on previous research made by Pellegrini et al. (2012). However, this paper proposes some changes to improve managing delays and support optimal automatic timetable planning. The constraints proposed by (Pellegrini et al., 2012) are explained in Table 7.

In Pellegrini et al. (2012), the $d_{t,tc}$ delay is used only if $bs_{tc} \neq bs_{tc}$ (the end of a block section). However, in our simulation environment (see Section 5), it is easier to interpret the local delay in every nominal track-circuit and manage it by constraints inserted into the MILP framework. In this case, there is no need for an additional mapping between all nominal track-circuits and those that correspond to a signal (end of a block section). The managing of delays in our approach is described by Eqs. (10)–(11).

$$d_{t,tc} = e_{t,tc} - e_{t,tc} - run_{tc} x_{t,tc} \quad (10)$$

$$\forall t \in T, tc \in TC_t : bs_{tc} \neq bs_{tc}$$

$$d_{t,tc} = 0 \forall t \in T, tc \in TC_t : bs_{tc} = bs_{tc} \quad (11)$$

Furthermore, based on constraint 8 prior delays were avoided by Pellegrini et al. (2012) and Pellegrini et al. (2015), which means that the trains entered the infrastructure exactly when they were scheduled. It can lead to an infeasible problem. For example, suppose two scheduled trains in the control area with entry times whose difference is less than the sum of the formation time, clearing time, and release time. In that case, the method proposed by Pellegrini et al. (2012) would fail. Neglecting the constraint 8 allows one to use the MILP framework for automatic timetable planning. Thus, the previous problem can be resolved by assigning prior delays outside of the control area to the trains. It means that the control area was assumed to have

signals at its entry points ($\forall tc \in TC: s_{tc} = tc_0, bs_{tc} \neq bs_{tc_0}$). It should be noted that this increases the computational cost of the algorithm because of the escalated number of possible solutions. The present solution has two main applications: real-time optimization and automatic timetable planning. In timetable planning applications, it is also beneficial to avoid primary delay if it is feasible. First, the primary delay must be defined and is performed by Eq. (12).

$$d_{t,tc0} = \sum_{\substack{tc \in TC_i: \\ p_{tc} = tc_0}} e_{t,tc} - init_t \quad (12)$$

Then, the computation of objective function is reformulated as in Eq. (13).

$$D_t = e_{t,tc\infty} - exit_t + d_{t,tc0} \quad (13)$$

The rolling stock configuration constraints must be applied for the trains which result from a joint, split, or turn-around of another train. These constraints are detailed in Table 8.

The capacity constraints consider the limits of the infrastructure and other railway traffic regulations detailed in Section 3. These constraints are explained in Table 9.

It is important to note that the value of M must be defined carefully. In the integer linear programming framework of Matlab, the integer condition has a tolerance. The tolerance should be less than the reciprocal of M 's scale. For example, let M be 10^4 . In this case, the integer tolerance in Matlab should be less than 10^{-4} (e.g., 10^{-5}).

Table 8 Constraints due to the change of rolling stock configuration

Constraint	Description
9	A minimum separation time must be granted between the departure of the train which results from the joint, split or turn-around of another train and the arrival of the corresponding train.
10	The train, from which the other train results, reserved the track-circuit where the rolling stock configuration change can take place until the new train started reserving this track-circuit.
11	Both trains use the route that contains the platform where the change of rolling stock configuration can be carried out (spatial coherence).

Table 9 Capacity constraints

Constraint	Description
12	The trains start to reserve the track-circuits which belong to the block section formation time unit before they entered them.
13	The trains reserve a track-circuit after the full rolling stock configuration had left the track-circuit until the release time elapses.
14–16	A real track-circuit can be reserved by only one train at the same time.

5 Simulation environment

Besides the infrastructure modeler and MILP-based rt-RTMP optimization algorithm, the Matlab toolchain also contains a simulation environment to evaluate and visualize the results. The simulation environment is suitable for real-time visualization of the trains on the infrastructure. Based on the simple train model described in Section 3, the simulator computes the estimated position of the trains in each timestamp with a defined simulation time resolution. This simulation time resolution is a user-defined parameter of the simulator. Besides the current location of the trains, the reserved track-circuits are also demonstrated by the edges highlighted with red. The trains are represented by gold rectangles. The train simulator GUI is shown in Fig. 7.

The comparison of the optimized solution with the traditional railway traffic management approaches is a crucial part of the evaluation. Thus, the simulator includes an in-built option to visualize the result of a global FIFO (First Input First Output) principle. In this method, the first train entering the control area should exit first, before any other trains that subsequently arrived in the control area. If there is a conflict between two trains, the train that entered the control area later than the other should be delayed by letting the other train to exit the control area first. This delay is assigned at a signal where collision avoidance can be ensured without giving delays to another train. In many cases, this principle can be applied only by primary delays. The primary delay means that the corresponding train could enter the control area later than it was scheduled. Hence, primary delays are applied to the trains outside of the control area.

There are three options implemented for visualization in Matlab: the initial problem without any traffic management rules (no delays are assigned to trains), the result of the FIFO principle, and the MILP-based solution of real-time railway traffic management problem. Among the running times and other train model-related parameters detailed in

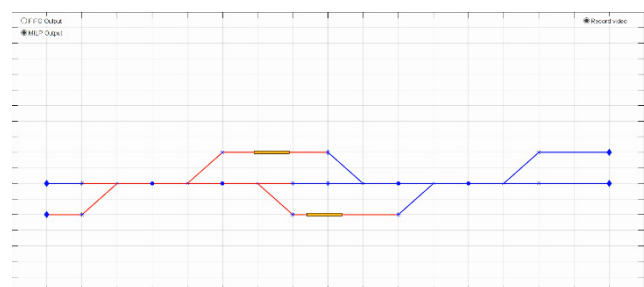


Fig. 7 The train simulator GUI

Section 3, the simulator takes other inputs to visualize the outcome of the traffic management. The selected route of each train and the delays assigned to them are given to the simulator. Based on the delays, a modified speed profile is also proposed for trains. The modified velocity of a train at a given nominal track-circuit is computed as in Eq. (14).

$$\hat{v}_{t,tc} = \frac{L_{tc}}{run_{t,tc} + d_{t,tc}} \quad (14)$$

Based on the proposed modified velocity, the train positions can be easily computed as a result of either of the solutions of the real-time railway traffic management problem.

6 Results

The results are evaluated based on the infrastructure model represented in Fig. 2. However, the platforms and the 8 routes corresponding to them have been neglected. Thus, the 13 real track-circuits and 12 routes generate 202 nominal track-circuits. The state optimized by the MILP algorithm consists of 1980 elements, including the total delay of each train. Furthermore, the evaluation has also been performed on an extended infrastructure model to investigate the effect of the increased number of decision points (signals). In the extended infrastructure model, a signal is assigned to each node of the graph model in every possible direction. Four trains are assigned to both infrastructure models with the same scheduled entry time (4 sec), desired velocity (100 km/h), and the total length of the rolling stock configuration (100 meters). The trains have different entry and exit locations. Train 1 travels from tc1 to tc13 and train 2 from tc2 to tc12. Train 3 and train 4 move in the opposite direction from tc13 to tc1 and tc12 to tc2, respectively.

The evaluation has been carried out in two aspects: the performance and the runtime result of the algorithms. In the evaluation of the performance, the value of the objective function is monitored.

The objective function values of the different solutions on both infrastructure models are summarized in Table 10.

The result of the four experiments is also represented by the speed-distance diagram of the trains in Figs. 8–11.

Table 10 Performance result of different rt-RTMP algorithms

Simple infrastructure model	
FIFO	53 sec
MILP	32.6 sec
Extended infrastructure model	
FIFO	53 sec
MILP	20.8 sec

Among the presence of trains at each track-circuit, the reservation time intervals are shown in these diagrams, as well. Note that the notations shown in Fig. 8 are the same as notations in Figs. 9–11 but we already have not displayed them there separately.

On the simple infrastructure model with fewer signals compared to the extended one, the objective value of the global FIFO principle regarding the summed delay of each train is 53 seconds, while the MILP-based optimization results in 32.6 seconds. It means that the optimization reduced the objective value by 38.5 %. Based on the speed-distance diagrams represented in Fig. 6 and Fig. 7, one can see that the traffic management was carried out in two main steps in the FIFO principle. In the first step, train 1 and train 2 travel from the left side to the right side of the control area. Due to collision avoidance, train 3 and train 4 cannot enter the control area; thus, a primary delay is assigned to these trains. However, MILP-based optimization enables more than two entering trains to the infrastructure simultaneously.

The result of the FIFO principle on the extended infrastructure model is shown in Fig. 10, which is very similar to the result of the simple model. The difference is only the track-circuit reservations due to different block sections, but the objective value is the same. However, the target value of the MILP algorithm is 20.8 seconds on the extended infrastructure model (shown in Fig. 11). It means a 60.75 % reduction of the summed delay of each train. Thus, with an increasing number of decision points in the control area, the efficiency of the MILP-based traffic flow management is growing compared to the FIFO principle.

The runtime is an essential aspect of real-time traffic management algorithms. Thus, the runtime of real-time traffic management optimization has been evaluated and compared to the result of the FIFO principle. The timetable planning application mode does not require real-time running. The runtime of the rt-RTMP algorithms was evaluated in Matlab IDE on Lenovo W530 ThinkPad with Intel Core i7-3520M 2.9 GHz processor (2 cores) and 16 GB memory. The algorithms were run 10 times and the average runtimes are summarized in Table 11. As was expected, the runtime of the MILP-based optimal traffic management algorithm is significantly greater than the runtime of the FIFO principle. Furthermore, the runtime of both algorithms increases with the increase of decision points.

However, the runtime of the FIFO algorithm seems to be more sensitive to the complexity of the infrastructure compared to the MILP. The runtime of the FIFO principle

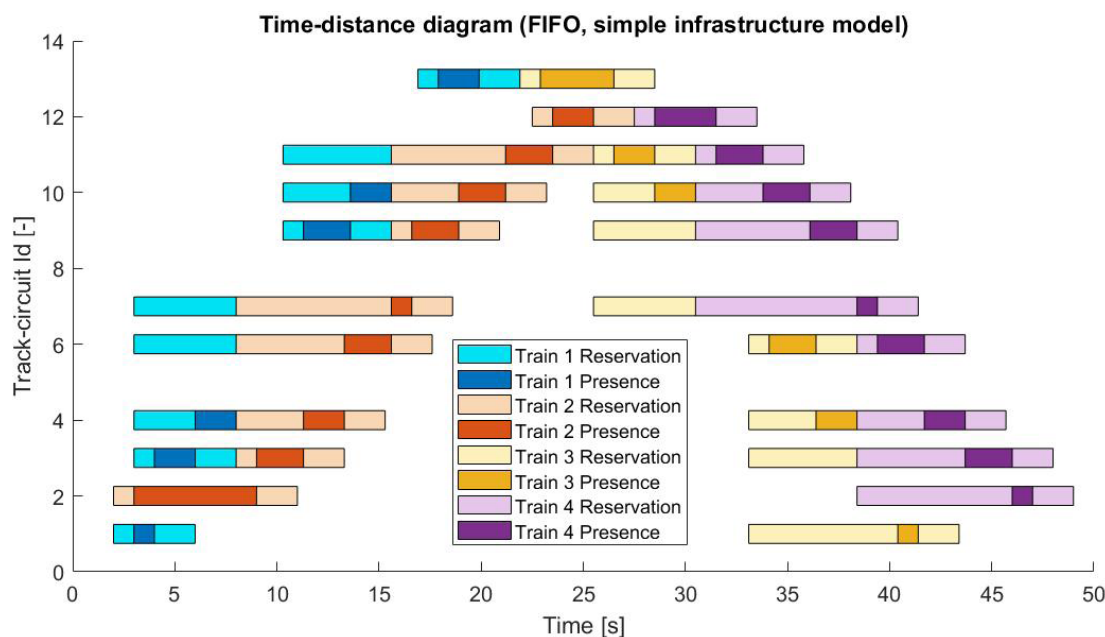


Fig. 8 The time-distance diagram of the FIFO solution on the simple infrastructure model

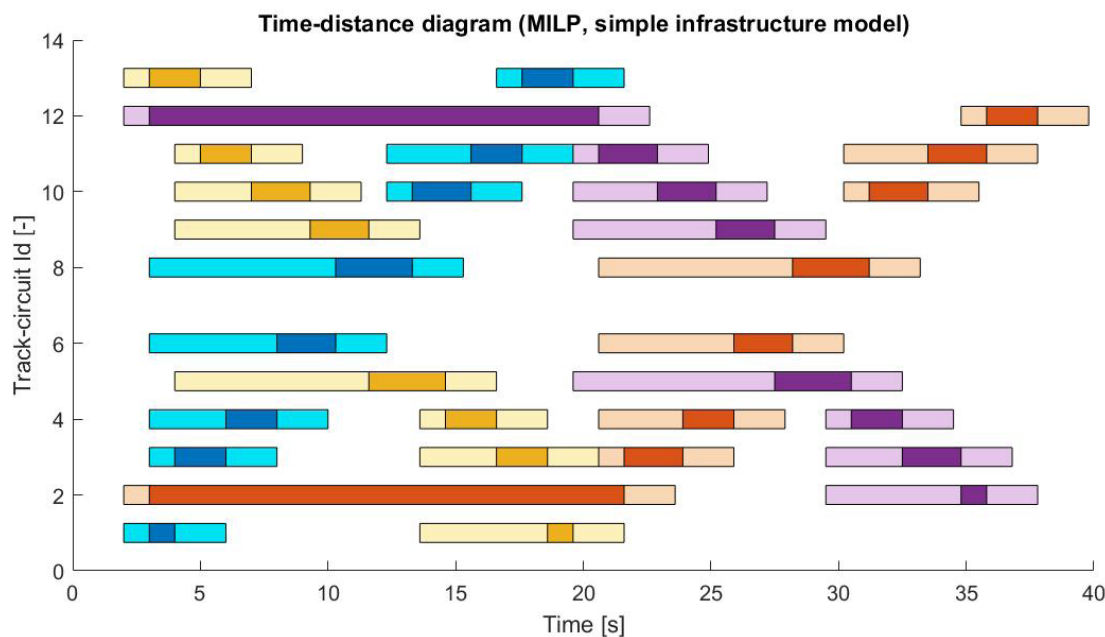


Fig. 9 The time-distance diagram of the MILP solution on the simple infrastructure model

is greater than 60 %, while the runtime of MILP increased about 20 % in extended infrastructure compared to the simple one.

7 Conclusion

In this paper, the process described by Pellegrini et al. (2012) and Pellegrini et al. (2015) has been applied with appropriate modifications to increasing the performance of the MILP algorithm in order to solve the rt-RTMP. To verify the result,

we used a simple case study. Based on the result, it can be concluded that the use of MILP can provide effective support to dispatchers, who generally apply the FIFO principle and rely on their experiences.

Our further research plans include checking our solution with larger infrastructure models, increasing the performance of this solution, and examining its practical applicability.

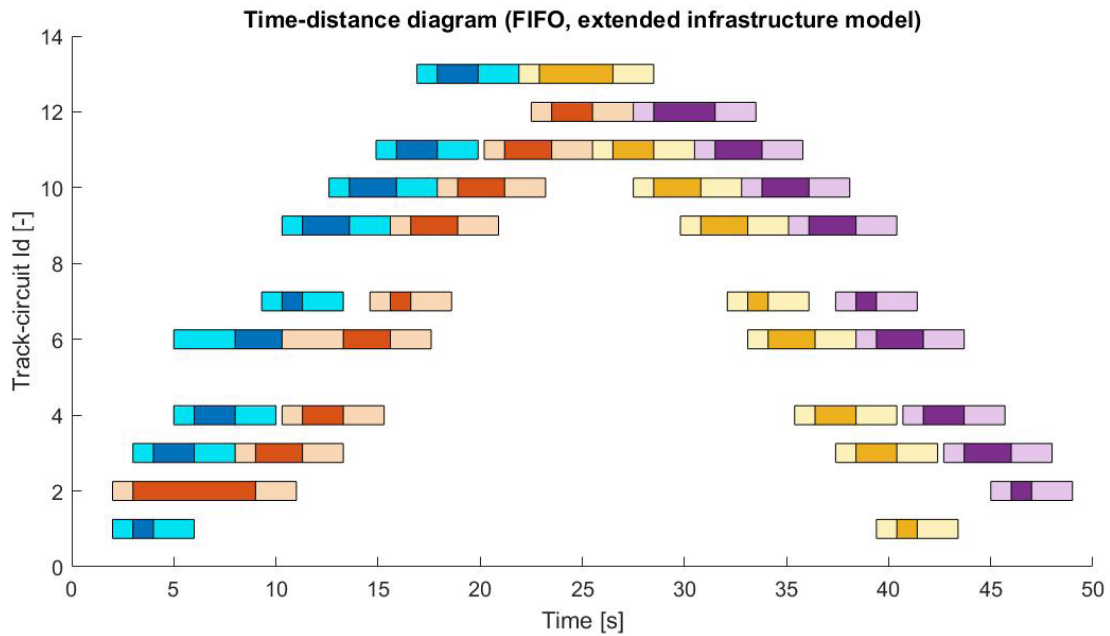


Fig. 10 The time-distance diagram of the FIFO solution on the simple infrastructure model

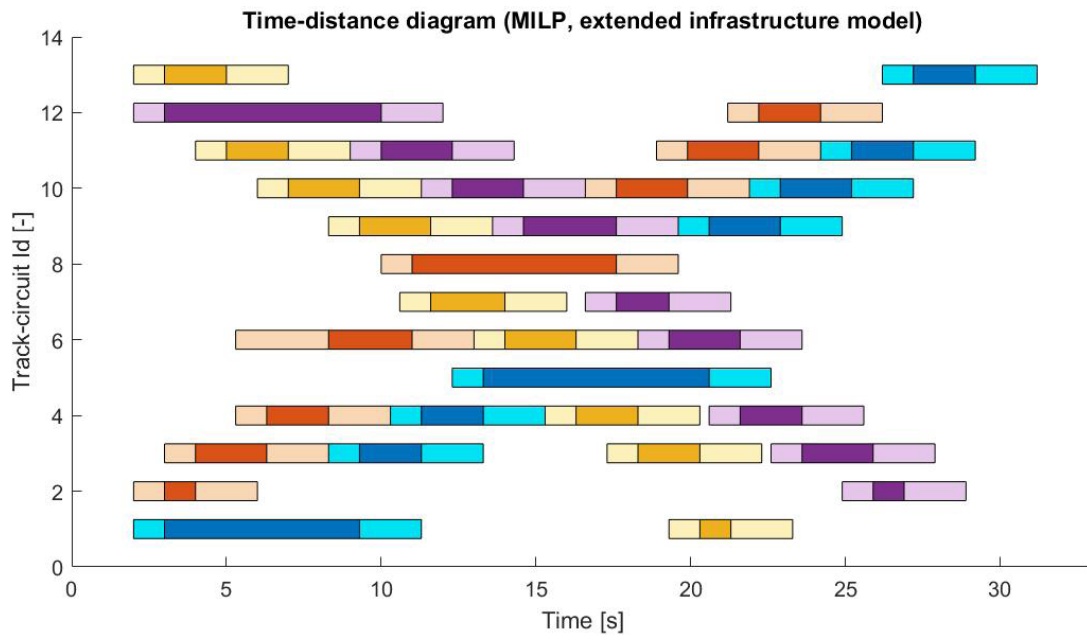


Fig. 11 The time-distance diagram of the MILP solution on the simple infrastructure model

Table 11 Runtime performance result of different rt-RTMP algorithms

Simple infrastructure model	
FIFO	0.03 sec
MILP	1.43 sec
Extended infrastructure model	
FIFO	0.05 sec
MILP	1.71 sec

Acknowledgement

This research is occurred in the project 2018-1.3.1-VKE-2018-00040, which is connected to Investigation of the possibility of energy efficient transport from point of view of interlocking systems.

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

References

- Abels, D., Jordi, J., Ostrowski, M., Schaub, T., Toletti, A., Wanko, P. (2019) "Train scheduling with hybrid ASP", In: Balduccini M., Lierler Y., Woltran S. (eds) Logic Programming and Nonmonotonic Reasoning. LPNMR 2019. Lecture Notes in Computer Science, Springer, Cham, pp. 3–17.
https://doi.org/10.1007/978-3-030-20528-7_1
- Dijkstra, E. W. (1959) "A note on two problems in connexion with graphs", *Numerische mathematik*, 1(1), pp. 269–271.
<https://doi.org/10.1007/BF01386390>
- European Commission (2019a) "COM(2019) 640 final Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions", European Commission, Brussels, Belgium. [online] Available at: https://eur-lex.europa.eu/resource.html?uri=cellar:b828d165-1c22-11ea-8c1f-01aa75ed71a1.0002.02/DOC_1&format=PDF [Accessed: 10 February 2021]
- European Commission (2019b) "COM(2019) 640 final ANNEX Annex to the Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions", European Commission, Brussels, Belgium. [online] Available at: https://eur-lex.europa.eu/resource.html?uri=cellar:b828d165-1c22-11ea-8c1f-01aa75ed71a1.0002.02/DOC_2&format=PDF [Accessed: 10 February 2021]
- European Commission (2021) "Seventh monitoring report on the development of the rail market under Article 15(4) of Directive 2012/34/EU of the European Parliament and of the Council, Package data and figures", European Commission, Brussels, Belgium Rep. SWD/2021/1 final/EC. [online] Available at: <https://ec.europa.eu/transport/sites/transport/files/2021-7th-rmms-report-package-data-and-figures.xlsx> [Accessed: 10 February 2021]
- Farooqi, H., Incremona, G. P., Colaneri, P. (2018) "Collaborative Eco-Drive of Railway Vehicles via Switched Nonlinear Model Predictive Control", *IFAC-PapersOnLine*, 51(30), pp. 626–631.
<https://doi.org/10.1016/j.ifacol.2018.11.225>
- Farooqi, H., Incremona, G. P., Colaneri, P. (2019) "Railway collaborative ecodrive via dissension based switching nonlinear model predictive control", *European Journal of Control*, 50, pp. 153–160.
<https://doi.org/10.1016/j.ejcon.2019.04.005>
- Kozen, D. C. (1992) "Depth-First and Breadth-First Search", In: *The Design and Analysis of Algorithms. Texts and Monographs in Computer Science*, Springer, New York, NY, USA, pp. 19–24.
https://doi.org/10.1007/978-1-4612-4400-4_4
- Novak, H., Vaak, M. (2018) "Energy-Efficient Train Traction Control on Complex Rail Configurations", In: *2018 26th Mediterranean Conference on Control and Automation (MED)*, Zadar, Croatia, pp. 1–9.
<https://doi.org/10.1109/med.2018.8442611>
- Pachl, J. (2014) "Timetable Design Principles", In: Albrecht, T., Hansen, I. A., Pachl, J. (eds.) *Railway Timetabling & Operations Analysis – Modelling – Optimisation – Simulation – Performance Evaluation*, Eurailpress, Hamburg, Germany, pp. 13–46.
- Pellegrini, P., Marlière, G., Rodriguez, J. (2012) "Real Time Railway Traffic Management Modeling Track-Circuits", In: *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'12)*, Ljubljana, Slovenia, pp. 23–34.
<https://doi.org/10.4230/OASIS.ATMOS.2012.23>
- Pellegrini, P., Marlière, G., Pesenti R., Rodriguez, J. (2015) "RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem", *IEEE Transactions on Intelligent Transportation Systems*, 16(5), pp. 2609–2619.
<https://doi.org/10.1109/TITS.2015.2414294>
- Toletti, A., De Martinis, V., Weidmann, U. (2016) "Energy savings in mixed rail traffic rescheduling: an RCG approach", In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, pp. 2430–2435.
<https://doi.org/10.1109/itsc.2016.7795947>